

PARTS OF SPEECH TAGGING USING HIDDEN MARKOV MODEL, MAXIMUM ENTROPY MODEL AND CONDITIONAL RANDOM FIELD

*Thesis submitted in partial fulfillment
of the requirement for the degree of*

Bachelor of Technology

in

Computer Science and Engineering

by

Anmol Anand

(Roll: 110cs0115)

Under the guidance of

Prof. S. K. Rath

NIT Rourkela



Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela-769 008, Orissa, India



Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela-769 008, India. www.nitrkl.ac.in

Dr. S. K. Rath

Professor

May 6, 2014

Certificate

This is to certify that the thesis entitled Parts of speech tagging using Hidden Markov Model, Maximum Entropy Model and Conditional Random Field by Anmol Anand for the partial fulfillment of the requirements for the award of Bachelor of Technology Degree in Computer Science and Engineering at the National Institute of Technology, Rourkela, is an authentic work carried out by him under my supervision and guidance. To the best of my knowledge, the matter embodied in the thesis has not been submitted to any other university / institute for the award of any Degree or Diploma.

Prof. S. K. Rath

Dept. of Computer Science and Engineering

National Institute of Technology Rourkela

Rourkela-769008

Acknowledgement

I am highly indebted to my guide Prof. S.K. Rath for giving me an opportunity to work under his able guidance. Like a true mentor, he motivated and inspired me through the entire duration of the work, without which this project could not have seen the light of the day.

I convey regards to all the other faculty members of the Department of Computer Science and Engineering, NIT Rourkela for their valuable guidance and advices at appropriate times. I would like to thank my friends for their help and assistance all through this project.

Last but not the least, I express my profound gratitude to the Almighty and my parents for their blessings and support without which this task could have never been accomplished.

Anmol Anand

110cs0115

BTech

Department of CSE, NIT Rourkela

Author's Declaration

I hereby certify that all the work contained in this report is done by me unless otherwise acknowledged. Also, all of my work has not been previously submitted for any academic degree. All sources of quoted information have been acknowledged by means of appropriate references.

Anmol Anand

110cs0115

B Tech

Department of CSE, NIT Rourkela

Abstract

Parts of Speech tagging assigns the suitable part of speech or in other words, the lexical category to every word in the sentence in Natural language. It is one of the essential tasks of Natural Language Processing. Parts of Speech tagging is the very first step following which various other processes as in chunking, parsing, named entity recognition etc. are performed.

An adaptation of various machine learning methods are applied namely Hidden Markov Model (HMM), Maximum Entropy Model(MEM) and Conditional Random Field(CRF) . For HMM models, we have used the suffix information for smoothing of the emission probabilities, while for ME model, the suffix information is used as features. Similar case for the CRF as that used by ME model.

The significant points brought about by thesis can be highlighted below:

- Use of Hidden Markov Model for Parts Of Speech tagging purpose. To create a sophisticated tagger using small set of training corpus , resources like a Dictionary is used that improves the overall accuracy of the tagger.
- Machine learning techniques have been introduced for acquiring discriminative approach. The Maximum Entropy Model and Conditional Random Field has been used for this task.

Keywords: Hidden Markov Model, Maximum Entropy Model, Conditional Random Field, POS tagger.

Contents

| | |
|---|-----------|
| Certificate | 2 |
| Acknowledgement | 3 |
| Author's Declaration | 4 |
| Abstract | 5 |
| 1 Introduction | 7 |
| 1.1 The Part-of-Speech Tagging problem | 8 |
| 1.2 Applications of POS tagging | 9 |
| 2 Tagging with Hidden Markov Model | 10 |
| 2.1 Hidden Markov Model | 10 |
| 2.2 Building HMM based model | 12 |
| 3 Tagging with Maximum Entropy Model | 13 |
| 3.1 Maximum Entropy Model | 13 |
| 3.1.1 Building the ME based model | 14 |
| 3.1.2 Features | 15 |
| 3.1.3 Training the model | 16 |
| 3.1.4 Decoding | 17 |

| | | |
|----------|--|-----------|
| 4 | Tagging with Conditional Random Fields | 18 |
| 4.1 | Conditional Random Field | 18 |
| 4.1.1 | Undirected Graphical Approach Models | 18 |
| 4.1.2 | Background | 19 |
| 4.1.3 | Estimating the parameters | 20 |
| 4.1.4 | Features | 20 |
| 5 | Analysis and Results | 21 |
| 5.1 | Parts of Speech tagging using Hidden Markov Model | 22 |
| 5.2 | Parts of Speech tagging using Maximum Entropy Model | 23 |
| 5.3 | 1 Parts of Speech tagging using Conditional Random Field | 25 |
| 6 | Conclusion | 27 |
| | Bibliography | 28 |

Chapter 1

Introduction

Parts of Speech tagging is the most basic task of language processing. A stream of text is taken as input and the corresponding parts of speech of every word is determined. A POS tagger is developed out of a set of linguistically motivated rules or a large training set tagged already. Such rules and training set are easily available for languages like English.

A Parts of Speech tagger plays an indispensable part of many Natural Language tasks such as Chunking, Parsing, Morphological analysis etc. A tagger facilitates in the process of annotated tagset creation. Some of the NLP related applications are Word sense disambiguation, Speech Recognition, Text to speech conversion, Machine translation.

Different approaches to POS tagging exists. Tagging tasks can be Supervised or Un-Supervised. Both are differentiated on the basis of degree of available training and tagged corpora. In case of Unsupervised POS model, previously annotated corpus is not required. Instead, advanced computational techniques are utilized to generate transformation rules, tagset etc. In case of Supervised POS model, previously annotated corpus is required which is used for training to get information about the tagset, tag sequence probabilities etc.

Recently, Machine learning techniques have been employed which takes annotated corpus to derive language knowledge for different NLP tasks. Using Machine learning techniques, taggers can be developed within short time, and the learning curve is very high. Lot of research has been carried out over POS derivation tasks. The algorithms vary from instance based learning to several graphical models. Taggers based on machine learning need to be trained with quite a lot of already tagged data. Now-a-days it is common to get a lot of tagged data for most of the languages, for the purpose of training the tagger. One explores the strength of Unsupervised, Semi-Supervised and Supervised learning mechanisms for POS tagger development.

NLP tasks are truly ambiguous and equivocal. Ambiguity may occur at distinctive levels of the Natural Language transforming assignment. There are numerous words that take various parts of speech tags. The right tag relies on the connection of utilization and the context in which it is used.

Sentences can have words with lots of POS ambiguity, and it is essential to get them determined before the sentence is understood. As an occurrence, the expression "dog" and "building" can be a verb or a noun, "on" might be a preposition, an adjective, an adverb; also, "tip" could be an adjective or a noun.

POS tagging is an assignment of suitable parts of speech labels to each word of the input sentence. Significantly, the POS tagging assignment determines ambiguity by selecting right tag from a conceivable tagset for an expression in a sentence. So, this issue could be seen as a classification task.

To be general, the stochastic meaning of the Parts Of Speech annotation might be expressed as. Given an arrangement of words $W=W_1 \dots W_n$, one needs to discover the suitable comparing arrangement of labels

$T = t_1..t_2...t_3..t_n$, drawn from a tagset [t], that fulfills the comparison:

$$S = \text{ARG MAXp}(t_1, t_2, t_3, t_n \mid w_1, w_2, w_3, w_n)$$

1.2 Applications of POS Tagging

Most of the natural language processing tasks need to remove parts of speech ambiguity. As so, it can be considered as the first step of language understanding. Further processes may include Parsing, Morphological Analysis, Chunking etc. Tagging is often a necessity for many applications as in Speech Analysis and Recognition, Machine translation, lexical analysis and information retrieval.

Natural language systems generally is composed of a set of interconnected pipelined tasks. Each of them is specific to a certain level of understanding and analysis of the text. Thus the development of POS tagger has a big impact over other of these pipelined tasks. Achieving a high degree of accuracy in this stage is very crucial as it may affect the further stages of natural language processing. Some of the applications of POS tagging are enumerated below:

1. Speech Recognition and synthesis, A remarkable amount of information is extracted about the word and its neighbours from its parts of speech. This information will be useful for the language model.
2. Machine Learning, POS tagging significantly affects the probability of translating a word from source language into target language.
3. Information extraction, on making a query to the expert system, a lot of information about the parts of speech can be retrieved. So, if one wants to search for files that contain 'building' as a verb, one can add additional information that removes the possibility of the word to be identified as a noun.

As said recently, POS tagging has been utilized within a few other requisition, for example, for handling high level syntactic and semantic preparation (noun phrase chunker), stylometry, lexicography and word sense.

Chapter 2

Tagging with Hidden Markov Model

In this part is depicted the HMM based calculation for Parts Of Speech tagging. Hidden Markov Model is a standout amongst the effectively utilized language model(1-gram..n-gram) for deriving labels which utilizes rare measure of data about the language, separated from simple context related data.

An HMM is a stochastic based construct which could be utilized to tackle the classification issues that have a state sequence form.

The model has a number of interconnected states connected by their transition probability. A transition probability is the probability that system moves from one state to another. A process begins in one of the states, and moves to another state, which is governed by the transition probability. An output symbol is emitted as the process moves from one state to the next. These are also known as the Observations. HMM basically outputs a sequence of symbols. The emitted symbol depends on the probability distribution of the particular state. But the exact sequence of states with respect to a typical observation sequence is not known (hidden).

2.1.1. DEFINITIONS

As per Rabiner, there are five components that need to be characterized in a HMM. The five tuples of a HMM are as follows:

1. The unique states (N) in a model. The distinct states are indicated as $S = \{s_1, s_2, \dots, s_N\}$. In the event of Speech labeling, N indicates the amount of labels in the set $\{t\}$ that is utilized by the framework. Each one tag in the set of tags is a fixed state in the model.
2. The different output symbols S in the model. The distinctive symbols are signified as $V = \{v_1, v_2, v_3, v_4, v_n\}$. For labeling, M is number of words present in the vocabulary set of the framework.
3. The transition probabilities which is denoted by $A = \{a_{ij}\}$. The likelihood a_{ij} , is the likelihood that the state moves from i to j in a move. In parts of speech labeling, states relate to labels, subsequently a_{ij} is the likelihood that the model moves from label t_i to t_j (where $t_i, t_j \in \{t\}$). To edge it in an alternate manner, a_{ij} is likelihood that t_j takes after t_i (that is $P(t_j | t_i)$). This likelihood is for the most part evaluated from the annotated training data during preparing.
4. The emitted output likelihood $B = \{b_j(k)\}$. Likelihood of $b_j(k)$ speaks to the likelihood that the k-th generated symbol will be generated when the system goes to state j . Thus for POS labeling,

it is the likelihood that the saying W_k is generated when the procedure is in state t_j (i.e. $P(W_k | t_j)$). It is likewise evaluated from the preparation corpus.

The starting sequence pattern is the likelihood that the system begins at state i . For POS labeling, this is the likelihood that the text will start with a specific label t .

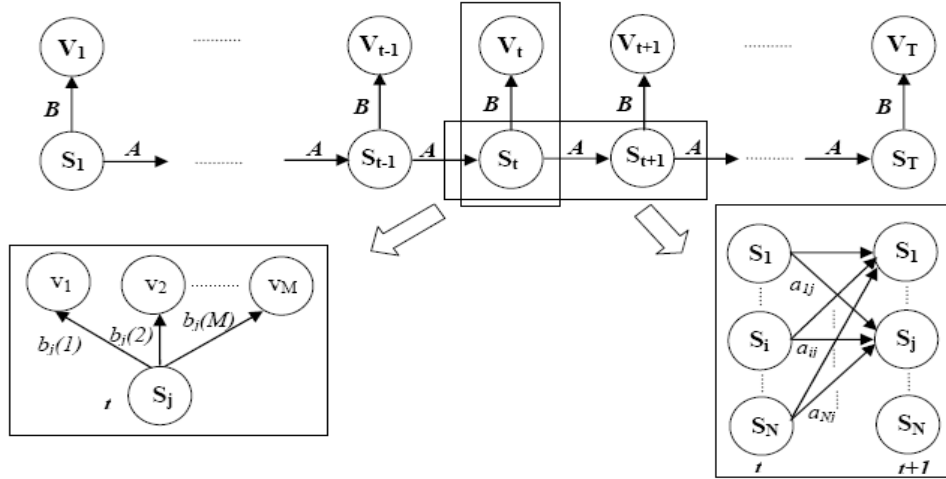


FIG 2.1 Sequence of states.

At the point when using an HMM to apply labeling, the point is to recognize the most likely probable label (states) grouping that produces the parts of speech of a text (the succession of output symbols). We calculate the sequence of tags S given a sentence W that maximizes $P(W | S)$. The Viterbi algorithm(dynamic programming standard) is utilized to figure out this maximum likelihood estimate. The calculation is examined in short in the subsequent sections.

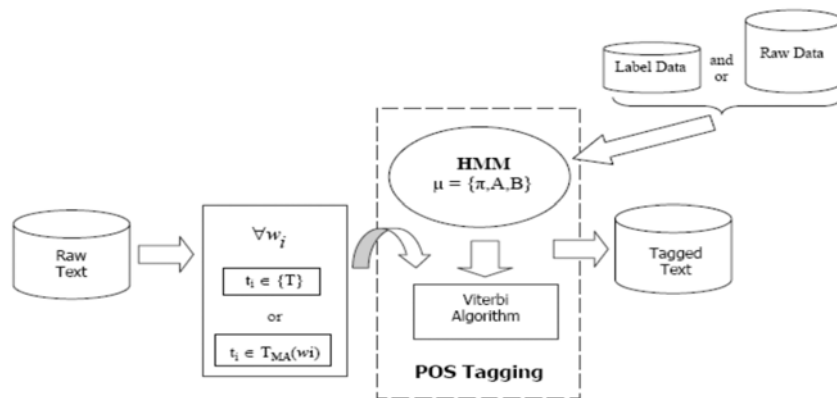


FIG 2.2 HMM architecture

An automatic POS labeling of characteristic natural text utilizing HMM is implemented. The framework has three segments, which are talked about above. Firstly, the framework needs some data

about the undertaking of disambiguation. Information can hail from numerous assets and could be set in different forms. This representation is known as the language model. The language model for Hidden Markov Model is spoken to have its parameters as (π, A, B) .

The point is to provide a highest estimation to the parameters (π, A, B) of the Hidden Markov Model utilizing the training set. The parameters of the model of the HMM are evaluated on the premise of the labeled text throughout supervised learning. The model parameters are again re-assessed utilizing the Baum-Welch calculation. Unlabelled content are utilized for re-estimation throughout semi-supervised learning. HMM Bigram model is utilized for the above implementation.

2.2.1 Models

There are a few approaches to speak to the HMM based model for automatic POS labeling as indicated by the way the learning is obtained. The HMM model utilizes the accompanying three parameters of data.

- (1) Probability of symbol emissions , that is the likelihood of specific label t_i , provided a specific word W_i , $P(W_i | t_i)$.
- (2) Probability of transition between states, i.e. the likelihood of a specific label relying upon the past labels, , $P(t_i | t_{i-1} t_{i-2} \dots t_{i-k})$.
- (3) Probability for beginning state, i.e. the likelihood of a specific tag as a starting of Markov approach. (V., 2007)

Chapter 3

Tagging with Maximum Entropy Model

In the previous section was discussed the HMM based stochastic language models for POS labeling. This includes a set of matching characteristics for the HMM tagger. Simple HMM fails to work as desired when sufficient labeled data is not provided to estimate the model parameters. Providing a rich set of features in HMM based tagger is not easy and complicates the smoothing. Under such circumstances, the Maximum Entropy model deals quite well and handle sparse data problems. So, a suitable way is to provide a rich combination of various features, which cannot be done in a natural way in HMM models.

In this part, we exhibit the work on ME based probabilistic calculation for POS labeling in English. The employment of distinctive Features and their viable execution in the Maximum Entropy Model is additionally presented.

3.1 ME Model

Maximum Entropy is an adaptable and flexible modeling system. This Model determines the probabilities based upon constraints. Upon the application of constraints the most probable sequence of tags is produced. These constraints are determined from the preparation information, keeping up connection between the history and probable Outcomes. Outcomes are the sets of permissible tags. Maximum Entropy model permits the estimation of $P(t | h)$ for given t in the space of aggregate conceivable outcomes T , for each 'h' chosen among the set of histories, H . A history in ME is the greater part of the required data that enables one to determine probabilities for the set of outcomes. In POS generation task, one can redefine it in term of discovering the likelihood of a label (t) associated to the word at any arbitrary index , in the test information as :

$$P(t | h_i) = p(t | \text{data inferred from the test information at index } i)$$

For a given collection of characteristics and the preparation corpus, the ME estimation methodology generate a model such that each feature f_i is connected with a type λ_i . This prompts the processing of the conditional likelihood as following:

$$P(t | h) = \frac{\prod_i \lambda_i f_i(h, t)}{Z_\lambda(h)}$$

$$Z_\lambda(h) = \sum_t \prod_i \lambda_i f_i(h, t)$$

As such, the above comparison indicates that the likelihood of the conclusion, provided the history, is the result of the weights of all features. These are then standardized/ normalized over the products for their outcome.

Maximum Entropy model generates a classified model for the characteristic features. The intention is to maximize entropy of the model. An attempt is made to reduce the amount of information that is carried out by the model. No extra assumption is made like the one taken in Hidden Markov Model. This is one of the distinctive features of the Maximum Entropy Model. Basically, a feature set is generated for the language model. One of the features may be 'The word ends in suffix with length not more than 4 letters'. Then we try to find the values of the feature sets for the training sample. The value is then optimized with the weight of the features, maximizing the entropy of the model. Based on the value, the proposed system will give the stochastic probability that the token comes within the given classes of token against which the language model was trained.

3.1.1 Building ME based model

The solution to the various parameters of the Maximum Entropy Model is generated using the Generalized Iterative Scaling (GIS) calculation. This approach is ensured to converge to a solution. A framework of the approach being applied to the probabilistic model is given underneath.

GIS: Provided a group of index features, and likewise the associated approximation of functions K_i , each iterating loop j makes refreshed estimation of the model parameters λ_i that compares the requirements better than the past one. Every iteration involved the following list of steps :

Process the expectation value of all the f_i under the new approximation of the likelihood function.

1. The expectancy of all the f_i within the new estimate of the likelihood function.

$$K_i^{(j)} = \sum_h \tilde{P}(h) \sum_t P_j(t|h) f_i(h,t)$$

2. Figure the real value of $K(j)$ and upgrade the λ_i as per the Formulae

$$\lambda_i^{(j+1)} = \lambda_i^{(j)} \cdot \frac{K_i}{K_i^{(j)}} \quad (\text{A., 1996})$$

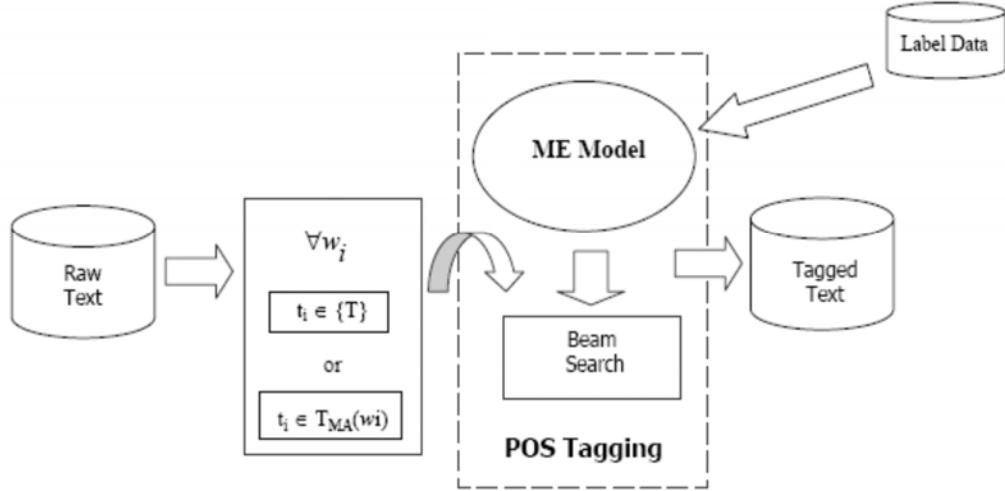


Figure 3.1: The ME based POS tagging architecture

3. Defining next approximation of the probability function on the new value of λ_i .

Continue iterating until convergence or close-convergence.

Specifically to the ME model, the language model segment is represented by the parameters of the model. As the case with HMM, there is an algorithm for the task of disambiguation, which chooses the maximum likelihood tag sequence to given word sentence. We utilize the beam search algorithm for this task.

3.2.1 Features

These are functions which possess binary value which associate labels with various elements of the context; a typical feature can be:

$$f(h, t) = \begin{cases} 1 & \text{if current_token}(h) = \text{Ami and } t = \text{PRP} \\ 0 & \text{otherwise} \end{cases}$$

Feature choice assumes a paramount importance in the ME framework. The principle features for the POS labeling undertaking are recognized focused around the diverse possible pattern of the word and label sequence. The features additionally keep prefix and additional suffix data for all words. The term prefix is a grouping of starting few characters of a word, which need not be semantically meaningful. The utilization of prefix data as features is discovered to be extremely powerful for exceptionally high inflected languages. We think about diverse mixtures from the accompanying set of features for identifying the best POS tags.

$$F = \{w_i, w_{i-1}, w_{i-2}, w_{i+1}, w_{i+2}, t_{i-1}, t_{i-2}, |pre| \leq 4, |suf| \leq 4\}$$

A representation of the various features is shown. The single line shows the whole feature set which consist of both the features, dynamic and static.

A pictorial representation of the different features is portrayed beneath. The single robust line represents entire feature set “F” which comprises of both static and dynamic features

Dotted lines specify those characteristics that are static and are obtained from text. The dashed line signifies the dynamic characteristics which are evaluated during execution time. The double solid line characterizes the output.

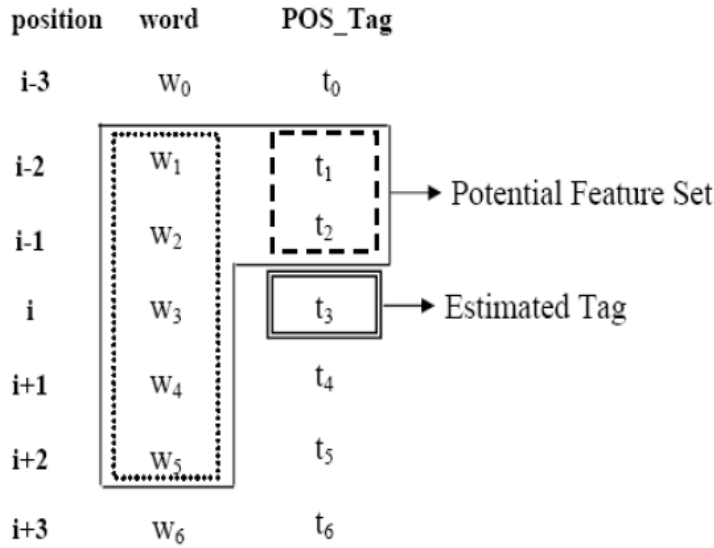


Figure 3.2 : Set of Features for MEM

3.2.2 Training the Model

Maximum Entropy model utilize a corpus hand-checked with the right POS names.

The system utilizes Generalized Iterative Scaling (GIS) to assemble the model, which is ensured to arrive to a solution. The technique of preparing the framework is pointed out beneath.

1. Define the training set, C.
2. Convert the preparation corpus into tokens
3. Create a record of features which can include lexical characteristics determined from the preparation corpus.

4. Creation of an active file posting each characteristic which initiates each pair $\langle h, t \rangle$ for $h \in C$ $t \in \{t\}$
4. Compute the ME weighting λ_i for each f_i utilizing the ME toolbox with the active file as input. (V., 2007)

3.2.3 Decoding

This issue of POS labeling might be formally expressed as following. Given an arrangement of words $w_1 \dots w_2 \dots w_3 \dots w_n$, we need to discover the sequence of labels $t_1 \dots t_2 \dots t_3 \dots t_n$, drawn from a set of labels T, which fulfils:

$$P(t_1 \dots t_n | w_1 \dots w_n) = \prod_{i=1 \dots n} P(t_i | h_i)$$

Where, h is the context/extra information of the word W_i . The Beam Search calculation is utilized to discover the most probable sequence given the sentence.

Let $W = \{w_1, \dots, w_n\}$ be an untagged text, and let S_{ij} be the j^{th} most elevated likelihood label grouping up to word W. The accompanying is the strategy for the beam search:

1. Generating the approximation of each one tag from the set $\{t\}$ for W, discover top N (beam size), set S_{1j} , $1 \leq j \leq N$, as need be.
2. Set $i = 2$.
 - (a) Set $j = 1$.
 - (b) Generating the likelihood of each one tag from the set $[t]$ for W, given $S_{(i-1)j}$ as the previous label connection, and annex each one tag to $s_{(i-1)j}$ to make a new sequence.
 - (c) Set $j = j+1$, and again repeat the (b) if $j \leq N$
3. Find N most elevated likelihood estimations created by the above loop iteration and set S_{ij} , $1 \leq j \leq N$, appropriately.
4. Set $i = i+1$, and again repeat from (a) if $j \leq N$
5. Output the most likelihood sequence S_{n1} . (V., 2007)

Chapter 4

Tagging with Conditional Random Fields

Observation shows that Maximum Entropy model shows improvement over the HMM model while little preparing information or small training data was provided. Anyways with a good measure of preparing information the execution of both the models are equivalent. Maximum Entropy models are manifestation of discriminative approach, which increases the likelihood estimation of the corpus set. CRF has a single exponential approach for deriving the joint likelihood of the whole sequence of states provided the output pattern. As the case with Maximum Entropy parameters, a CRF based technique can likewise manage various covering features. A CRF is an extremely adaptable strategy which manages the inadequate information issue well. Within it, a characteristic mixture of various set of features could be effectively used, which isn't possible commonly in HMM.

4.1 Conditional Random Fields

Hidden Markov Models are more generally utilized for Parts of speech sequence tagging. HMMs are generative models, which try to maximize the joint probability distribution $P(X,Y)$ where X and Y are random variables, which represent the observation sequence and the corresponding label sequence. This joint probability distribution will generate the observation depending on the state or tag at that time. This assumption surely works for a simple data set. However this cannot be considered appropriate approach for tagging in general. Observation sequence must depend on multiple features and dependencies.

One approach to fulfill the requirement is to utilize a model that characterizes maximum likelihood $p(y | x)$ over marked arrangements given a specific observation grouping x , as opposed to a joint likelihood estimation over observation sequence and labels. Conditional based models are utilized to mark an unknown pattern of observation, choosing the sequence label which boosts the conditional likelihood.

4.1.1. Undirected Graphical Approach Models

Conditional Random Fields are essentially Markov Field or undirected graphical model, which are conditioned on X . X is a random variable that represents the observation sequence. $G = (V,E)$ is an undirected graph with V as the node such that $v \in V$, corresponding to every random variable that represent an element Y_v of Y . To state it, if every random variable Y_v obeys the Markov property with respect to G , then (Y,X) is a conditional Random Field.

A simple first order chain is illustrated below.

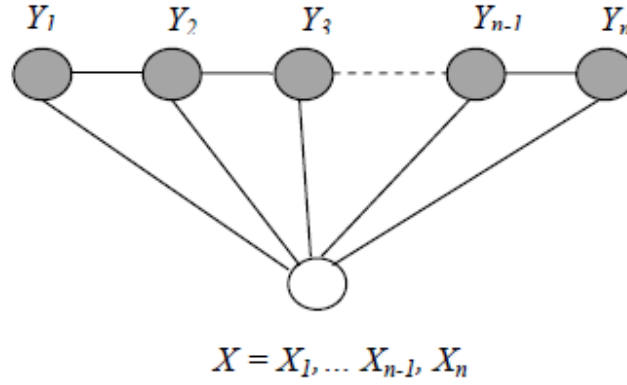


FIG 4.1 Chain-structured CRF (A Graphical Structure)

4.1.2 Background

Lafferty and others characterize the likelihood of a specific name grouping y provided the observation pattern x to be a standardized result of the essential features, each of the structure

$$\exp\left(\sum_j \lambda_j t_j(y_{i-1}, y_i, x, i) + \sum_k \mu_k s_k(y_i, x, i)\right)$$

Where $t_j(y_{i-1}, y_i, x, i)$ is a transition functionality featuring the sequence of observation in the label pattern. $s_k(y_i, x, i)$ is a state function of tag at a particular position i and sequence of observation. λ and μ are the model parameters to be assessed from the preparation information. (Lafferty J., 2001)

As the case with the ME model, when characterizing characteristic capacities in CRF model, we build a genuine set of features $b(x, i)$ of the observations to represent a few qualities of the preparation information.

Each of the feature functions undertake the quality of one of the absolute observation feature $b(x, i)$ if the current state or present and past states assume some specific qualities. Subsequently every feature functions are true esteemed in actuality.

Case in point, considering about the accompanying feature functions:

$$t_j(y_{i-1}, y_i, x, i) = \begin{cases} b(x, i) & \text{if } y_{i-1} = NN \text{ and } y_i = PP \\ 0 & \text{otherwise} \end{cases}$$

$$F_j(y, x) = \sum_{i=1}^n f_j(y_{i-1}, y_i, x, i)$$

Where

$$F_j(y, x) = \sum_{i=1}^n f_j(y_{i-1}, y_i, x, i)$$

4.1.3 Estimating the Parameters

Accepting preparation information $\{ (x_k, y_k) \}$ are independently spread, the product of comparison over training corpus, as a function of the parameter λ , is known as the probability, meant by $p(y_k | x_k, \lambda)$.

Increasing likelihood probability training picks parameters such that the log of the probability, also known as the log-probability, is expanded. For a CRF, the log-probability is given by

$$L(\lambda) = \sum_k \left[\log \frac{1}{Z(x^{(k)})} + \sum_j \lambda_j F_j(y^{(k)}, x^{(k)}) \right]$$

4.1.4 Features

Features are functions which have a value 0 or 1, which sets a tag with different components of the pattern; which was previously discussed.

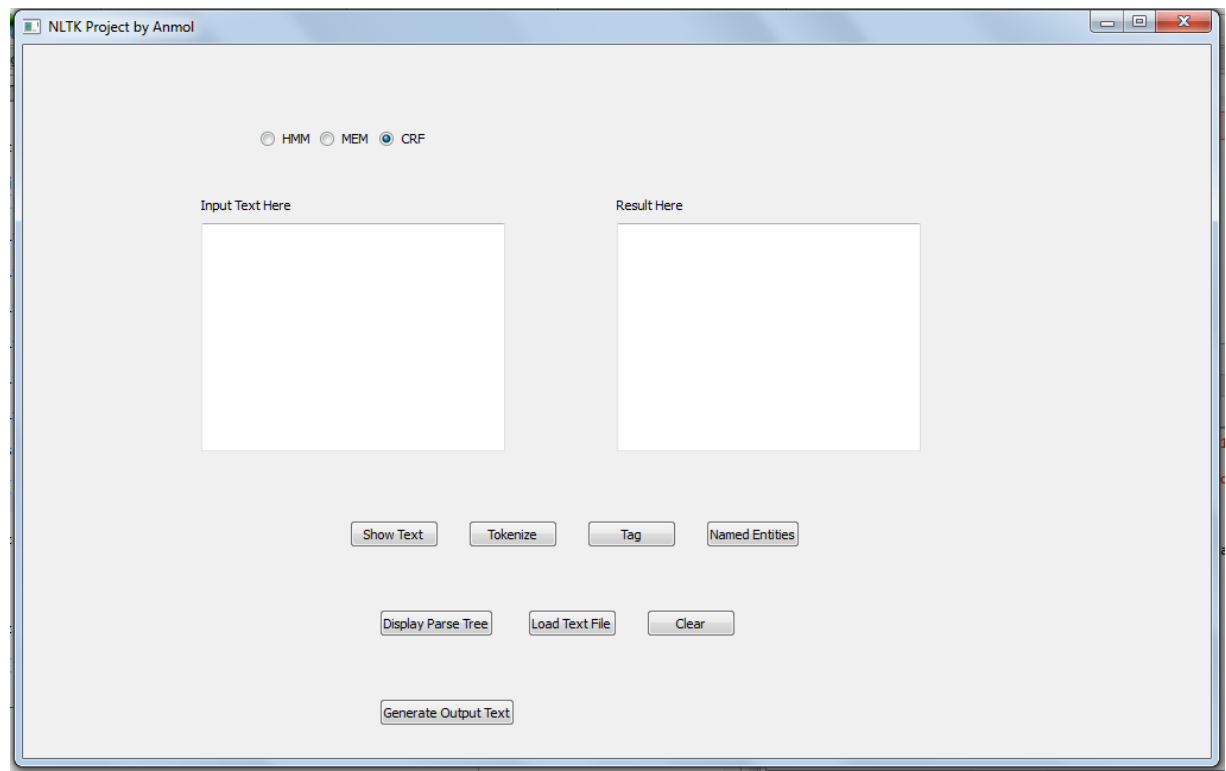
Feature selection assumes a pivotal part in the CRF structure. The principle features for the POS labeling activity have been distinguished based around the diverse possible blend of accessible word and label setting. The features likewise incorporate prefix for all words. The term prefix is a succession of starting first few characters of an expression, which does not mean a phonetically genuine prefix. The utilization of prefix data works well for exceptionally bent inflected words. We acknowledged diverse mix from the accompanying set for assessing the best features for POS labeling undertaking:

$$F = \{ w_i, w_{i-1}, w_{i-2}, w_{i+1}, w_{i+2}, t_{i-1}, t_{i-2}, |pre| \leq 4, |suf| \leq 4 \} \quad (A., 1996)$$

From the observational perception one finds in the Maximum Entropy based POS labeling. Model an extremely basic characteristic of the current word, past label and prefix gives the best bring about the new experimental conditions.

Chapter 5

Analysis and Results



5.1 The toolbox

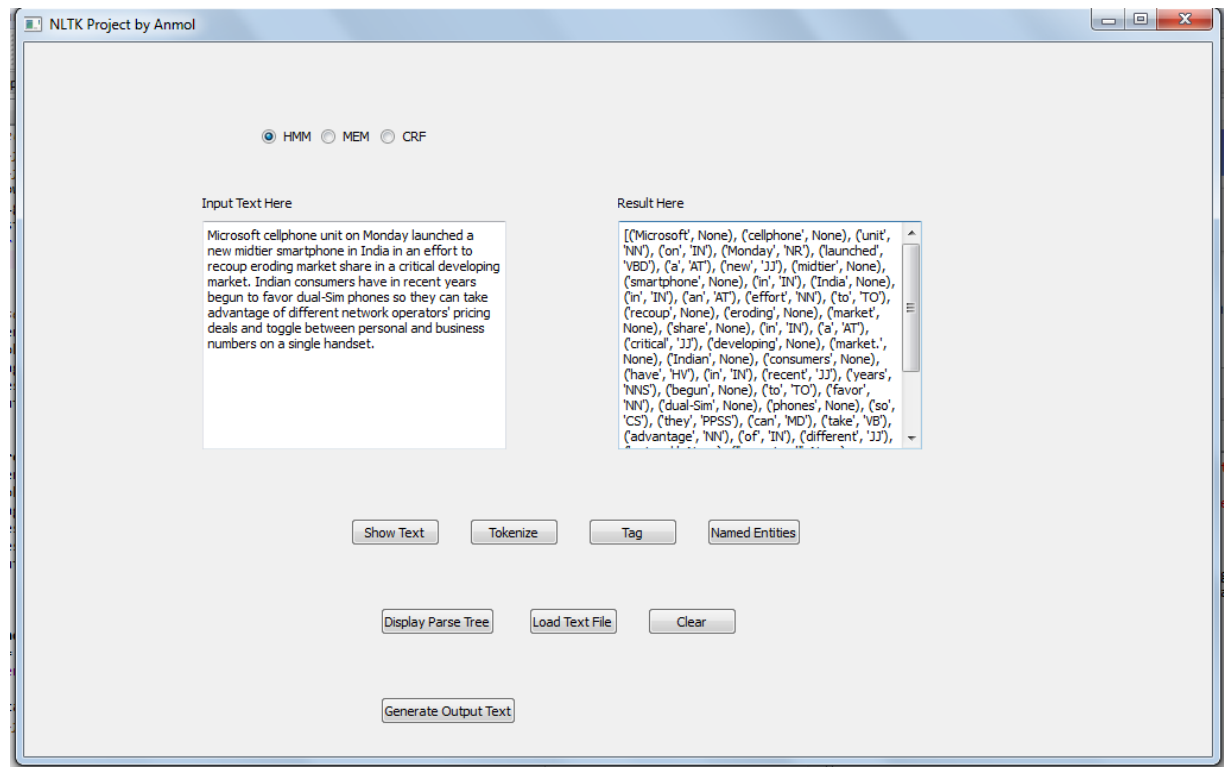
The toolbox above implements all the three models, the HMM model, the ME model and the CRF. This toolbox is designed in Python.

Following are the features of it :

- Tokenize : Convert the sentence into tokens
- Tag : Tags the words according to their parts of speech
- Named Entities : Derives the various named entities in the text
- Parse Tree generation : Generates a semantic parse tree.

PARTS OF SPEECH TAGGING USING HIDDEN MARKOV MODEL

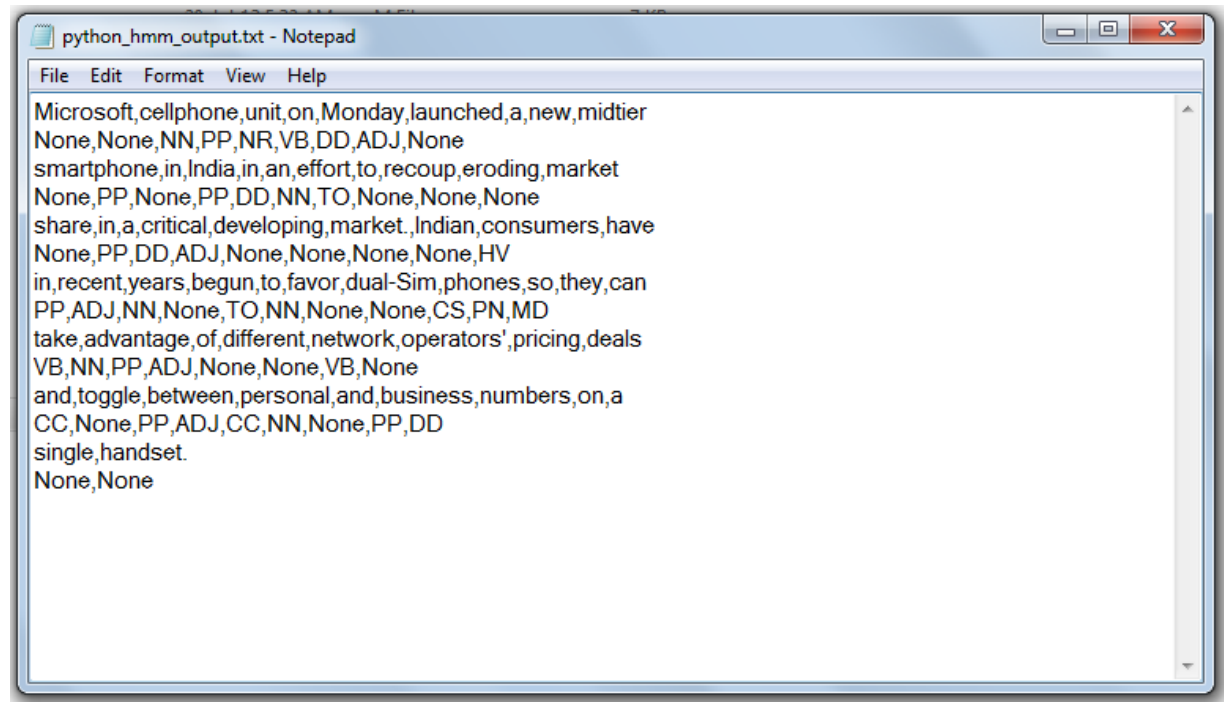
Below is the implementation of the HMM model. As can be seen, in the left text box, input text is entered. Once an appropriate action/ operation is performed, the result is obtained and shown in the right box.



5.2 HMM Tagger

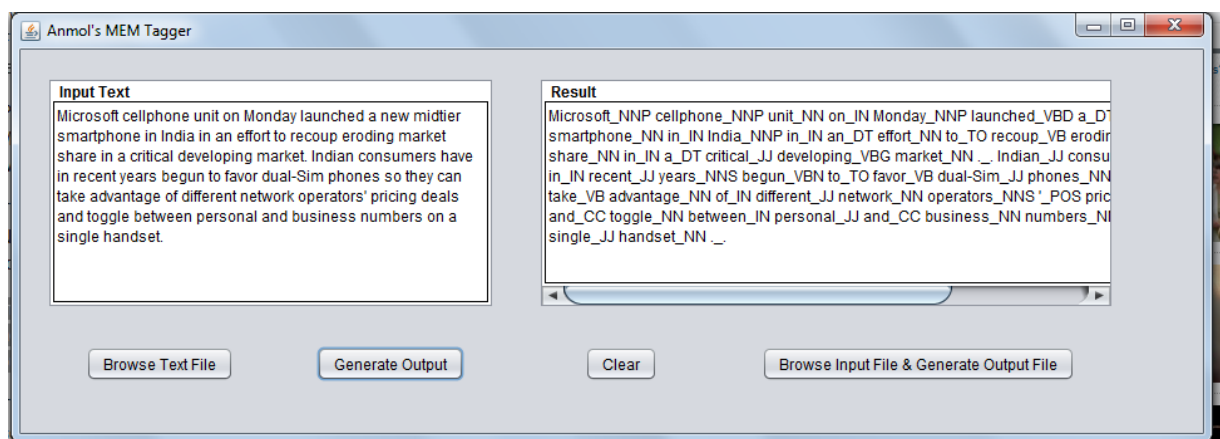
The sentence can be entered in two forms, either in sentence form or by taking the whole file as the input. After the tokenization process, the respective tokens are fed into the tagger. The tagger then generates the respective parts of speech of the words.

Below can be seen the output file for the HMM implementation. The parts of speech tagger writes the tags with the words in an output file as shown. This file keeps information of the various parts of speech generated against the words of the sentence/input file.



5.3 HMM Output File

PARTS OF SPEECH TAGGING USING MAXIMUM ENTROPY MODEL

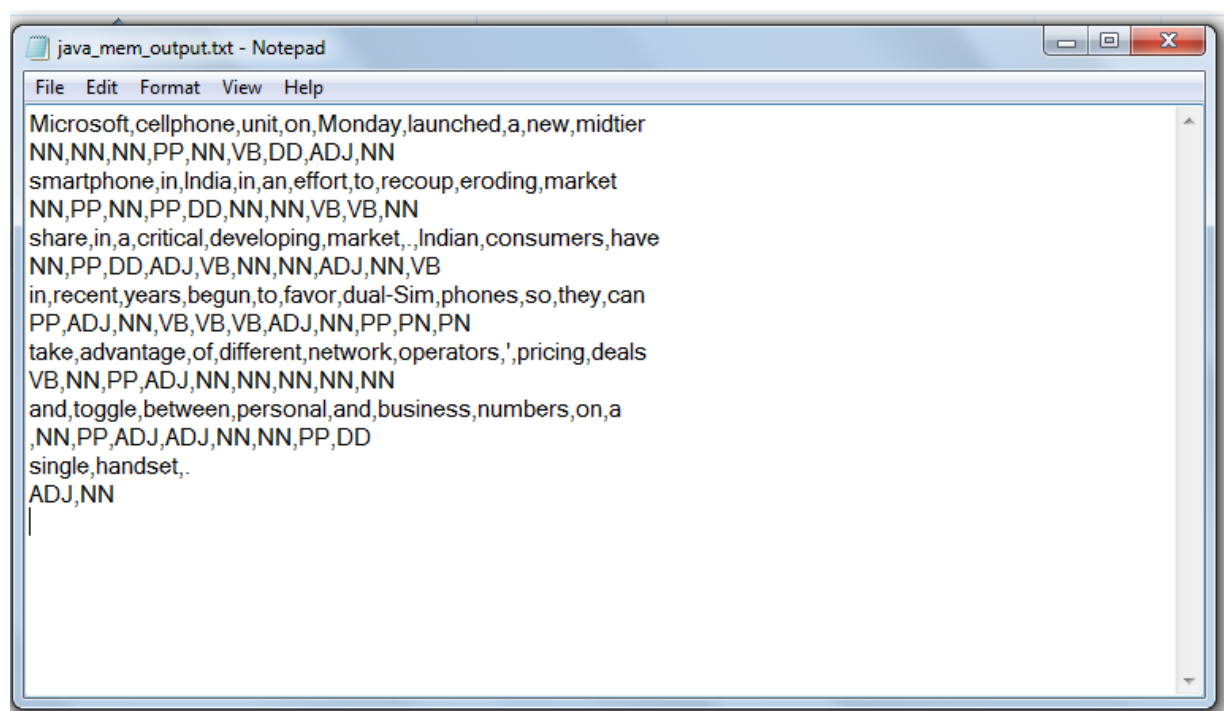


5.4 MEM Tagger

This is the proposed tool that implements Maximum Entropy Model for parts of speech tagging.

The sentence can be entered in two forms, either in sentence form or by taking the file as a whole as the input. After the tokenization process, the respective tokens are fed into the tagger. The tagger then generates the respective parts of speech of the words.

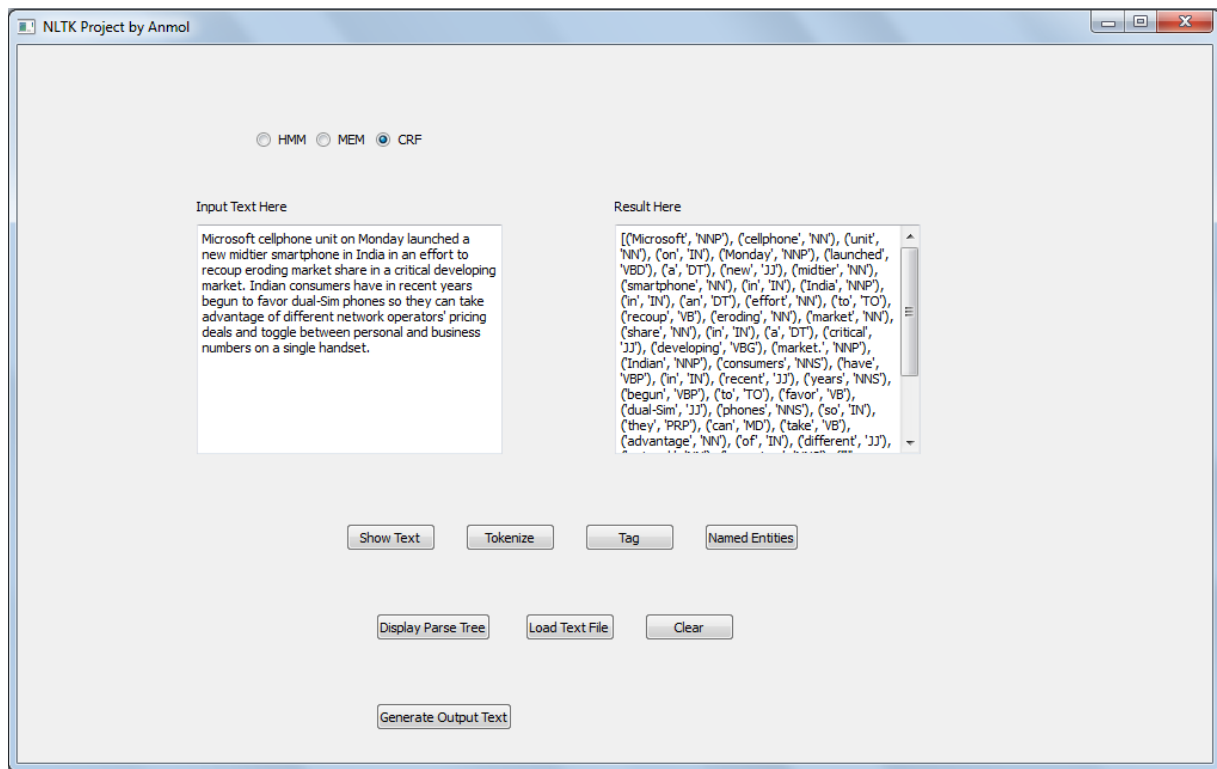
Below can be seen the output file for the MEM implementation. The parts of speech tagger writes the tags with the words in an output file as shown. This file keeps information of the various parts of speech generated against the words of the sentence/input file.



```
java_mem_output.txt - Notepad
File Edit Format View Help
Microsoft,cellphone,unit,on,Monday,launched,a,new,midtier
NN,NN,NN,PP,NN,VB,DD,ADJ,NN
smartphone,in,India,in,an,effort,to,recoup,eroding,market
NN,PP,NN,PP,DD,NN,NN,VB,VB,NN
share,in,a,critical,developing,market,,Indian,consumers,have
NN,PP,DD,ADJ,VB,NN,NN,ADJ,NN,VB
in,recent,years,begun,to,favor,dual-Sim,phones,so,they,can
PP,ADJ,NN,VB,VB,VB,ADJ,NN,PP,PN,PN
take,advantage,of,different,network,operators,',pricing,deals
VB,NN,PP,ADJ,NN,NN,NN,NN,NN
and,toggle,between,personal,and,business,numbers,on,a
,NN,PP,ADJ,ADJ,NN,NN,PP,DD
single,handset,.
ADJ,NN
|
```

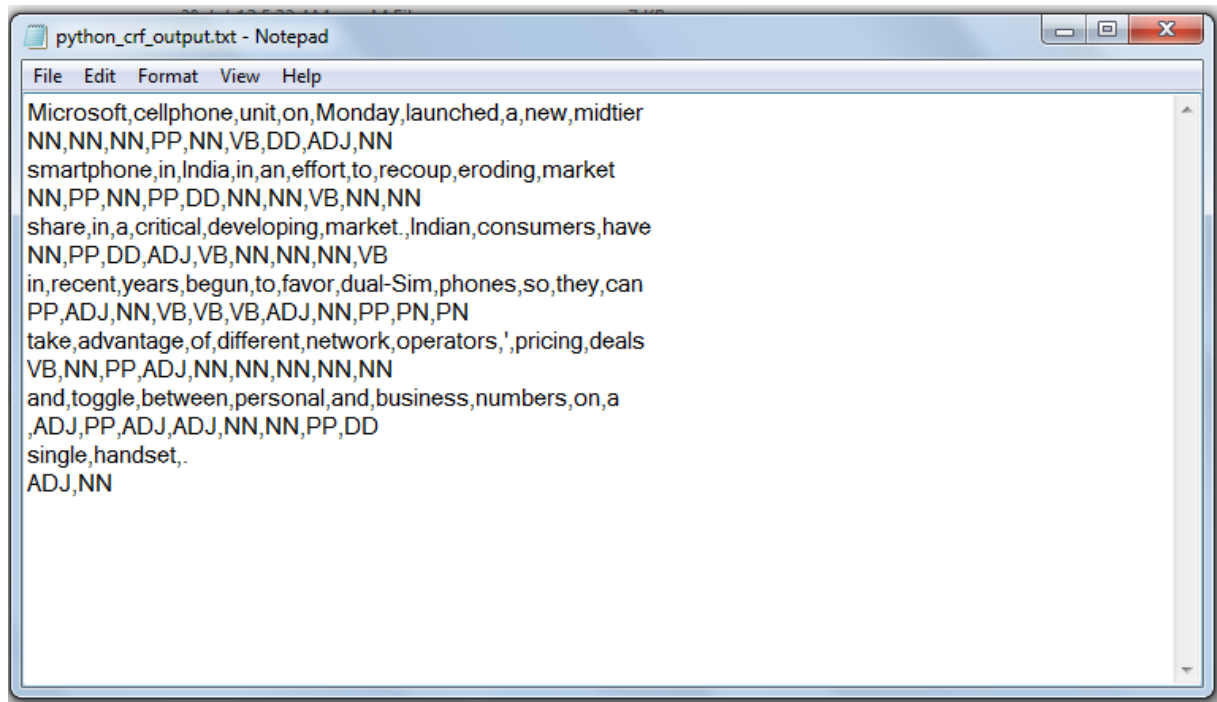
5.5 MEM Output File

PARTS OF SPEECH TAGGING BY CONDITIONAL RANDOM FIELD



5.6 The CRF Tagger

The sentence can be entered in two forms, either in sentence form or by taking a file as a whole as the input. After the tokenization process, the respective tokens are fed into the tagger. The tagger then generates the respective parts of speech of the words.



5.7 CRF Output File

CRF output file is shown above. The parts of speech tagger writes the tags with the words in an output file as shown. This file keeps information of the various parts of speech generated against the words of the sentence/input file.

| Model | Precision | Recall | F-Measure |
|-------|-----------|--------|-----------|
| HMM | 0.5365 | 0.4859 | 0.4989 |
| MEM | 0.8852 | 0.8473 | 0.8752 |
| CRF | 0.9016 | 0.8649 | 0.8922 |

These were the results obtained when the Models were made to run on the Brown Corpus which consisted of 500 lines of text.

A part of the corpus was used for training the taggers.

Clearly, CRF taggers were quite accurate followed by ME Tagger. HMM performed quite poorly on the brown corpus.

Chapter 6

Conclusions

In this work, we have examined various Natural Language Processing techniques and applied machine learning techniques, like the Hidden Markov Model (HMM), the Maximum Entropy Model (MEM) and the Conditional Random Fields (CRF). We have used the Brown Corpus for training and testing.

The HMM model discussed in this thesis are quite straightforward and simple, but efficient for deriving the most probable tags for the input sentence, even when very small training set is provided. The best accuracy is however achieved for the supervised bigram HMM model, taking also the morphological features of the language, and additional data for rare words.

In spite of the fact that HMM performs better for parts of speech disambiguation task, it uses the local features (previous one or two tags, current word) for POS tagging. Using the local features might not work well for a morphologically rich and relatively free order language, while Maximum Entropy Model is quite adaptable to such features and exhibits better usability of features.

A Maximum Entropy Model is utilized for automatic POS tagging. Maximum Entropy based tagger is superior to others in terms when morphological limitation is taken into consideration. HMM tries to find the most probable tag sequence for a given set of words, rather the Maximum Entropy takes into account the characteristic features of the word for determining the conditional likelihood of the tags for the sentence.

Talking of the CRF model, they perform better compared to the two mentioned models. Same characteristic feature set was used for CRF as the one used by the Maximum Entropy model. CRFs try to maximize the joint probability of the entire sequence of states provided, the observation sequence. CRFs stand out powerful because of its rich, diverse and overlapping feature set.

Speaking in broader terms, all the models discussed in the thesis are quite straightforward and proficient for automatic parts of speech tagging of Natural Language, when the training data is not present in sufficient amount.

Bibliography

- [1] Arulmozhi P., Rao. R. K. Sobha L., (2006). A Hybrid POS Tagger for a relatively Free Word Order Language. In proceedings of the Modeling and Shallow Parsing of Indian Language (MSPIL), Bombay 79-85.
- [2] Abney S., (1997). Part-of-speech tagging and partial parsing. In Ken Church, Steve Young, and GernitBloothoof, editors, Corpus-Based Methods in Language and Speech. Kluwer, Dordrecht.
- [3] Brill E., (1995a). Transformation-based error-driven learning and Natural Language Processing: A case study in part-of-speech tagging. Computational Linguistics, 21(4). 543-565.
- [4] Lafferty J., McCallum A. and Pereira F., (2001). Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In Proceedings of the Eighteenth International Conference on Machine Learning. 282-289.
- [5] Ratnaparkhi A., (1996). A maximum entropy part-of-speech tagger. In Proceedings of the Empirical Methods in Natural Language Processing Conference. 133-142.
- [6] Brown P., Della Pietra V., de Souza P., Lai J. and Mercer R., (1992). Class-based n-gram Models of Natural Language. Computational Linguistic, 18(4): 467-480.
- [7] Darroch J. and Ratcliff D., (1972). Generalized Iterative Scaling for log-linear models, Ann. Math. Statistics, 43 , 1470-1480.
- [8] Ronald Rosenfeld. (1994), Adaptive Statistical Language Modeling: A Maximum Entropy Approach. Carnegie Mellon University, Ph.D. Thesis.
- [9] Shrivastav M., Melz R., Singh S., Gupta K. and Bhattacharyya P., (2006). Conditional Random Field Based POS Tagger for Hindi. In Proceedings of the MSPIL, Bombay,. 63-68.
- [10] Wallach H. M., (2002). Efficient training of conditional random fields. Master's thesis, University of Edinburgh, 2002
- [11] Dasgupta S. and Ng V., (2007). Unsupervised Part-of-Speech Acquisition from Resource-Scarce Languages. In Proceedings of the Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, Prague. 218-227.